

Specification Amendments:

Please amend the specification as indicated:

Please replace the "CoPending Applications" section, beginning at line 1, on page 1 of the specification, with the following amended section:

Copending Applications

Q' A copending application exists having U.S. App. No. 09/489,682serial number ~~XXXXXX,XXX~~, entitled "Method and System for Accessing Packetized Elementary Stream Data", having at least one inventor in common, and the same filing date as the present application.

A copending application exists having U.S. App. No. 09/491,120serial number ~~XXXXXX,XXX~~, entitled "Method and Apparatus for Accessing Transport Stream Data", having at least one inventor in common, and the same filing date as the present application.

A copending application exists having U.S. App. No. 09/490,350serial number ~~XXXXXX,XXX~~, entitled "Method and System for Receiving and Framing Packetized Elementary Stream Data", having at least one inventor in common, and the same filing date as the present application.

A copending application exists having U.S. App. No. 09/491,119serial number ~~XXXXXX,XXX~~, entitled "Method for Synchronizing to a Data Stream", having at least one inventor in common, and the same filing date as the present application.

A copending application exists having U.S. App. No. 09/491,122serial number ~~XX/XXX,XXX~~, entitled "Method and Apparatus for Handling Private Data From Transport Stream Packets", having at least one inventor in, and the same filing date as the present application.

A copending application exists having U.S. App. No. 09/490,207serial number ~~XXXXXX,XXX~~, entitled "Method and System for Retrieving Adaptation Field Data

Associated with a Transport Packet", having at least one inventor in common, and the same filing date as the present application.

A copending application exists having U.S. App. No. 09/489,681 serial number ~~XXXXXX,XXX~~, entitled "Method for Displaying Data ", having at least one inventor in common, and the same filing date as the present application.

A copending application exists having U.S. App. No. 09/491,124 serial number ~~XXXXXX,XXX~~, entitled "System For Simulating The Parsing Of A Transport Data Stream", having at least one inventor in common, and the same filing date as the present application.

Please replace the 4th paragraph, beginning at line 19, on page 2 of the specification, with the following amended paragraph:

The transport stream (TS) specified by the MPEG-2 standard, offers a high degree of robustness for noisy channels, and can be used to carry multiple programs, such as multiple TV services. The transport stream is based on a 188 byte long packet suited for hardware error correction and processing schemes. The use of a robust protocol, such as the transport stream, allows for reception over noisy environments such as terrestrial and satellite transmissions. Even in these environments it is possible to obtain fast program access, channel hopping, and synchronization between multiple elementary streams carried within the packetized elementary streams which are subdivided into transport packets.

Please replace the first full paragraph, beginning at line 3, on page 6 of the specification, with the following amended paragraph:

One problem associated with the flexibility of the MPEG-2 standard is that once the transport stream is received, demodulated, and decrypted, the

Q³ resulting data stream can still have a ~~variety variations~~variety of variations which need be known before the data stream can be properly utilized. For example, the MPEG-2 specification does not indicate a specific set of control signals to be provided with the transport stream, how received data and control signals are qualified, nor the precise format of the actual data transmitted. As a result, implementations of set top boxes require specific service provider information. Specific service provider information results in an incompatibility among transport streams schemes provided by different service providers or cable operators. As a result, chip sets are designed and dedicated to support specific service provider's set top boxes.

Please replace the 2nd full paragraph, beginning at line 14, on page 11 of the specification, with the following amended paragraph:

Q⁴ The TPP 420 is connected to ~~the bus 450~~the bus 405, and receives the IN SYNC and PACKET START signals. Parsing of a TSP (packet) by the TPP 420 is enabled when the IN SYNC signal and the PACKET START signals are asserted indicating the beginning of a ~~new packet~~new packet. During parsing of the header portion of a packet the PID number is obtained. Based upon the value of the PID number, registers are updated, and a determination is made whether the TSP is to be saved, further processed, or discarded.

Please replace the 3rd full paragraph, beginning at line 20, on page 11 of the specification, with the following amended paragraph:

Q⁵ When it is determined to save the packet, the TPP 420 asserts the signal labeled ~~EN_TP~~TPP_DEN which is received by the Buffer Controller 460. Based upon this enable signal, the Buffer controller 460 retrieves the packet data and stores it in a predefined memory location.

Please replace the last paragraph, beginning at line 26, on page 11 of the specification, with the following amended paragraph:

Q6 When it is determined to further process the packet by one of the other parsers 450 or 430, the TPP 420 asserts one of their respective enable signals. For example, if it is determined that the packet contains video data, the TPP 420 will assert the signal labeled PESP-ENEN_PESP, likewise, if it is determined that the packet contains adaptation field data, the TPP 420 will assert the signal labeled AFP EN. Based upon these signal being active, the respective parser will further process the packed data.

Please replace the 2nd full paragraph, beginning at line 8, on page 12 of the specification, with the following amended paragraph:

Q7 When it is determined to save the video payload, the ~~PESP-420~~PESP 430 asserts the signal labeled EN_PESP_PESP_EN which is received by the Buffer Controller 460. Based upon this enable signal, the Buffer controller 460 retrieves the packet data and stores it in a predefined location of video memory.

Please replace the 1st paragraph, beginning at line 1, on page 13 of the specification, with the following amended paragraph:

Q8 Figure 7 illustrates another embodiment of a TS core in accordance with the present invention. The TS core of Figure 7 includes framer 710, TPP 720, AFP 750, PESP 730, ~~buffer controller 750~~buffer controller 760, and registers 780.

Please replace the 2nd paragraph, beginning at line 4, on page 13 of the specification, with the following amended paragraph:

Q⁹

The registers 780 are analogous to registers described with reference to ~~Figure 7~~ Figure 5.

Please replace the 4th paragraph, beginning at line 11, on page 13 of the specification, with the following amended paragraph:

Q¹⁰

The FRAMER DATA and FRAMER DEN signals are provided to each of the parsers of Figure 7, and the Buffer controller 760. The TPP parser 720 receives the header information of new packets when the framer 710 asserts an IN SYNC signal ~~and an~~ and a PACKET START signal. The combination of these signals, when asserted, indicate that the present FRAMER DATA is part of the packet header. As a result, the TPP 720 receives the FRAMER DATA from the data bus for parsing.

Please replace the 2nd full paragraph, beginning at line 15, on page 15 of the specification, with the following amended paragraph:

Q¹¹

The transport stream data and control signals can be received either from a direct broadcast or through a specific service provider. The signals actually received by the framer 710 can vary depending on the specific network interface module (NIM) provider of direct broadcast implementation. At a minimum, TCLOCK, and TDATA are needed. The TCLOCK and TDATA signals contain the basic information necessary to retrieve ~~is information~~ this information. While Figure 8 illustrates separate TDATA and TCLOCK signal, it is possible to provide the data and clock as an integrated signal, whereby the clock signal would be extracted from the received data.

Please replace the 2nd full paragraph, beginning at line 11, on page 17 of the specification, with the following amended paragraph:

Q12 While the timing diagram of Figure 9 does not explicitly show bits of TDATA being received serially, it should be understood that for each byte representation of TDATA in Figure 9, 8 individual data bits can be received, qualified by eight TCLOCK pulses, to form the bytes illustrated. When TDATA is received in a bit-by-bit manner, without a TSTART signal, there is no knowledge as to which of the bits being received represents the first bit of a byte, where by "first bit" it is meant the first bit received when the device is turned on and started latching the data. Likewise, the same is true for the first byte, let alone which byte represents the first byte of the frame. The state diagram of Figure 10 is a state diagram describing synchronizing the decoder core 700 of Figure 7 to the transport stream being received.

Please replace the 5th paragraph, beginning at line 20, on page 26 of the specification, with the following amended paragraph:

Q13 Figure 16 ~~illustrates~~ illustrates a more detailed view of the ~~TPP 620~~ TPP 720. ~~TPP 620~~ TPP 720 further includes storage locations 721, a counter controller 722, register controller 723, video PID location 724, and adaptation field start detect circuit 725.

Please replace the 2nd full paragraph, beginning at line 11, on page 30 of the specification, with the following amended paragraph:

Q14 In another implementation, control module 755 is controlled by the EnableParsing field (not shown in Figure 21) of the video control registers of Figure 18. The EnableParsing field is a one bit field, which when deasserted, prevents further parsing of the video packet by the video PESP. Therefore,

Q¹⁴ when the EnableParsing field is negated, the header of the video packet would not be parsed, and therefore, the packet would be discarded. The counter controller can be controlled directly from the EnableParsing bit of the video control registers, or indirectly where the VIDEO signal is disabled by the ~~TPP 620~~TPP 720 in response to the EnableParsing bit being deasserted.

Please replace the 3rd full paragraph, beginning at line 15, on page 31 of the specification, with the following amended paragraph:

Q¹⁵ The video PESP parser is bufferless in that no local buffers are used to store the payload data for access by other parts of the system. The prior art parsers stored the parsed data in large buffers locally, which were then capable of being accessed by system components by requesting access ~~to the local bus~~to the local bus. The bufferless parsers of the present invention do not store data locally for access by the system. Instead, parsed data to be buffered is transmitted to the buffer controller 460, which buffers data in system or video memory.

Please replace the last paragraph, beginning at line 22, on page 31 of the specification, with the following amended paragraph:

Q¹⁶ Figure 22 illustrates a method associated with a video PESP parser. At step 216, the PESP has received an indication that a video packet is ready to be parsed. The notification can be directly from the TPP, by a polling mechanism, or other type interrupt. ~~Step 16~~Step 216 determines whether parsing of the video stream is enabled. This can be determined based upon the field labeled EnableParsing of the video control registers of Figure 18. When parsing of the video packet is not enabled, a specific action will occur. One action would be to perform no further processing of the packet, as illustrated. In another implementation, the packet would be automatically

Q¹⁶ stored without further parsing, perhaps with the packet header field. When parsing of the video packet is enabled, the flow proceeds to step 217.

Please replace the last paragraph, beginning at line 25, on page 32 of the specification, with the following amended paragraph:

Q¹⁷ At step 223, the packet data is sent to the buffer controller for storage, as discussed with reference to ~~Figure 25~~ Figure 24.

Please replace the 3rd full paragraph, beginning at line 20, on page 35 of the specification, with the following amended paragraph:

Q¹⁸ In operation, the System FIFO controller 466 provides an interface between the Parsers of Figures 5 and 7 and the FIFO 462. The controller 460 allows filtered packet data to be received and stored in the FIFO 462. Once stored in the FIFO 462, the System HBI controller 463 requests access to the ~~video memory 471~~ system memory 472 through the controller 468. The controller 468 may include a system bus controller, a memory controller, or a combination of a memory/system bus controller. Generally, the controller 468 will control access to other system resources as well.

Please replace the 3rd full paragraph, beginning at line 12, on page 37 of the specification, with the following amended paragraph:

Q¹⁹ Figure 27 illustrates a method in accordance with the present invention describing the operation of the system HBI controller 463 of Figure 26. The flow is also applicable to the video HBI ~~controller 463~~ controller 483. At step 801, a determination is made whether there is data stored in the FIFO 462. If not, flow remains at step 801 until data is present, otherwise, the flow

proceeds to step 810. At step 810, the buffer to which the data is to be stored is identified. The destination buffer is identified when matching and crossing the PID number, or identifier, to the buffer number in the transport demultiplexer register 465. The buffer can be identified by accessing the allocation table, or by receiving a buffer index from the transport parser or other portion of the transport core.

Please replace the last paragraph, beginning at line 21, on page 37 of the specification, with the following amended paragraph:

At step 802, a determination is made whether the identified buffer is full, or otherwise not capable of ~~receives~~receiving additional data. If the buffer is not capable of receiving additional data, the flow loops back to step 802 through step 811, which implements a delay. Note the delay of step 811 may be a fixed delay, as result of polling to determine if the buffer is full, or the delay of step 811 may be variable, such as where the delay is based an interrupt which indicates when the buffer is available. Once the desired buffer is no longer full, flow proceeds to step 803.

Please replace the 3rd paragraph, beginning at line 16, on page 38 of the specification, with the following amended paragraph:

The buffer implementation described provides an advantage over the prior art in that moving the buffers in to system and video memory associated with an external system, such as a general purpose computer, allows for bufferless parsers. As a result, the system and video resources do not have to wait to access buffers local to the parsers. The performance improvements using bufferless parsers ~~has been~~has been observed by the present inventors to be up to 40% over the prior art. In addition, by allowing for parsing of the PES data, it is possible to limit the amount of bandwidth used to store unused

Q²¹ data. One skilled in the art will recognize the present invention has been described with reference to a specific embodiment, and that other implementations and variations of the present invention would be anticipated. For example, when a TSP is "sent" from the TP to the PESP or the buffer controller, it is to be understood that not necessarily all of the header information need be sent. In fact, it would generally be necessary for only the PID associated with the packet be forwarded. In addition, the location and implementation of the register sets and functionality described herein can be partitioned in ways other than the specific implementations described.

Please replace the 2nd full paragraph, beginning at line 8, on page 39 of the specification, with the following amended paragraph:

Q²² The AFP 750 illustrated in Figure 28 includes adaptation control counter 782, latch 785, register logic 786, and storage and register locations 781, 783, and 784. In operation, the adaptation control counter 782 receives signals on connections labeled AF START, FRAMER DEN, and FRAME DATA. The connection labeled AF START receives signals from the Transport Packet Parser 720, and indicates the beginning of the transport packet's adaptation field. The connection labeled FRAMER DEN receives signals from the ~~Framer 714~~ Framer 710, and indicates when each new byte of data is available on the FRAMER DATA bus. Based upon the received signals, the adaptation control counter 782 provides the control signals necessary to parse specific field information from data received on the FRAME DATA bus.

Please replace the 2nd full paragraph, beginning at line 4, on page 43 of the specification, with the following amended paragraph:

Q²³

Figure 31 illustrates control registers associated with ~~register set 784~~register set 780 that control operations associated with the Adaptation Field Parser 750.

Please replace the last paragraph, beginning at line 25, on page 45 of the specification, with the following amended paragraph:

Q²⁴

The AF Data Type Code storage ~~location 741~~location 742 stores the specific eight-bit type indicator associated with the adaptation field private data. The PES Data Type Code storage location 743 stores the specific the eight-bit type indicator associated with the PESP private data. The Stuffing Code storage location 744 stores the specific eight-bit stuffing code which is used to pad private data packet to insure the private data packet always ends on a double word boundary. The AFP Data Latch 745 is used to store the actual private data from the adaptation field parser to be provided to the buffer controller 760. Similarly, the PESP Data Latch is used to store the actual private data from the PESP parser is to be provided to the buffer controller 760. The fixed length indicator code 747 stores the fixed length value associated with the PESP parser private data. In the specific example, the PESP parser private data will always be 16 bytes of data, or 0x10 hexadecimal.

Please replace the 1st full paragraph, beginning at line 3, on page 53 of the specification, with the following amended paragraph:

Q²⁵

At step 333, the splice flag interrupt is enabled. The Splice Flag Interrupt Enable Bit is asserted in order to allow for the recognition of the splice in point. From step

Q²⁵ 333, the flow proceeds to step 302 of ~~Figure~~ Figure 35. Note that in another embodiment of the present invention, a determination step could be made at the beginning of the flow of Figure 38 as to whether the new PID is associated with a desirable program. If not, an alternate flow ignoring the PID, or using a dummy or alternate PID, could be used. For example, this feature could be used eliminate viewing commercials or other program types.

Please replace the 3rd full paragraph, beginning at line 19, on page 53 of the specification, with the following amended paragraph:

Q²⁶ At step 337, a determination is made whether or not the in-splice point indicator is valid. The in-splice point indicator is validated by determining whether or not random access flag register is set along with discontinuity flag register. The random access flag register, and discontinuity flag register, should both be set because the first packet of a new data stream will indicate the current packet is capable of being randomly ~~accesses~~ accessed by the system, and since no previous packets are associated with the PES stream the discontinuity flag should be set.

Please replace the 2nd full paragraph, beginning at line 6, on page 57 of the specification, with the following amended paragraph:

Q²⁷ At ~~step 912~~ step 913 of Figure 39, a determination is made whether the verification routine was successful. If so, the flow proceeds to step 914, if not, the flow proceeds to step 915.

Please replace the 5th full paragraph, beginning at line 15, on page 57 of the specification, with the following amended paragraph:

Q²⁸ Figure 42 illustrates a flow diagram that can that increments the transport stream characteristic in such a manner that all possible combinations are covered. By executing this routine, a successful increment will be indicated for all values, except for when BIT_ORDER variable is equal to MSB, and all other characteristics are equal to one. This state indicates that all possibilities have been tested, and an unsuccessful return occurs.

Please replace the paragraph beginning at line 1, on page 59 of the specification with the following amended paragraph:

Q²⁹ The input output (I/O) ~~adapter 1026~~ adapter 1022 is further connected to, and controls, disk drives 1047, printer 1045, removable storage devices 1046, as well as other standard and proprietary I/O devices.